# Week 6: Pointers and Memory Management

This week’s topic may be one of the more difficult ones in C++. So, we’ll do some reading, then some pointer exercises, and suggest a project for your own time.

## Background

We will start by going over any questions you may have about memory management. Then, we will do some exercises. Finally, we will talk about a data structure called a linked list. Wikipedia is always a great resource: <https://en.wikipedia.org/wiki/Linked_list>.

## Exercises

Here are some code segments. What do you think is the output? Why?

### Pointer Basics

int main() {

int i = 51, \*p1;

p1 = i;

cout << p1 << endl;

return 0;

}

int main() {

int i = 51, \*p1;

p1 = \*i;

cout << p1 << endl;

return 0;

}

int main() {

int i = 51, \*p1;

p1 = &i;

cout << p1 << endl;

return 0;

}

int main() {

int i= 51, \*p1;

p1 = &i;

cout << \*p1 << endl;

return 0;

}

### Mystery Code 1

#include <iostream>

using namespace std;

int main() {

int arr[] = { 1 , 2 , 3 , 4 , 5 , 6 , 7, 8 , 9 , 10 };

for ( int i=0 ; i<5 ; ++i ) {

int \*p1 = arr+i;

int \*p2 = arr+10-1-i;

int p3 = \*p2;

\*p2 = \*p1;

\*p1 = p3;

}

cout << "arr is:";

for ( int i=0 ; i<10 ; ++i ) {

cout << " " << arr[ i ];

}

cout << endl;

return 0;

}

### Mystery Code 2

#include <iostream>

using namespace std;

int main() {

int arr1[] = { 1 , 3 , 5 , 7 , 9 };

int arr2[] = { 2 , 4 , 6 , 8 , 10 };

int arr3[ 10 ];

int \*p1 = arr1 , \*p2 = arr2 , \*p3 = arr3;

while( p1 != arr1 + 5 || p2 != arr2 + 5 ) {

if ( p1 == arr1 + 5 ) {

\*p3++ = \*p2++;

}

else if ( p2 == arr2 + 5 ) {

\*p3++ = \*p1++;

}

else {

\*p3++ = (\*p1 < \*p2) ? \*p1++ : \*p2++;

}

}

cout << "Contents of arr3:";

for ( int i=0 ; i<10 ; ++i ) {

cout << " " << arr3[ i ];

}

cout << endl;

}

## Reading

Please visit this URL: <https://github.com/mjchao/Data-Structures-and-Algorithms/blob/master/LinkedList.h> and read through the code. We’ll have a discussion about any parts that are unclear. Don’t worry too much about the templates and class inheritance – just pay attention to the functions and how they interact, and how they use pointers.

## Outside Project

Consider practicing implementing your own LinkedList. Ignore the templating for now, and see if you can create a LinkedList of int that allows you to add and remove integers to/from the list at any given location.